UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/226,939	01/08/1999	JOHN K. VINCENT	346872000500	8916

24024 7590 05/19/2004

CALFEE HALTER & GRISWOLD, LLP
800 SUPERIOR AVENUE
SUITE 1400
CLEVELAND, OH 44114

EXAMINER	
LY, ANH	

ART UNIT	PAPER NUMBER
2172	25

DATE MAILED: 05/19/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/226,939	VINCENT ET AL.	
	Examiner	Art Unit	
	Anh Ly	2172	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 April 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *9-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *9-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed on 4/13/2004 have been fully considered but they are not persuasive.

Applicants' argued that, "Applicants find no disclosure by Morgenstern of at least a dependency graphing process that 'queries a database for dependency information.'" (Page 10, lines 12-13).

Morgenstern of 5,970,490 teaches development tools from CAD being applied to database schema on the relational and object oriented databases (col. 3, lines 8-32). A schematic structure graph from schema analyzers creates logical structure diagrams representing the dependency information of for databases (col. 11, lines 8-54). The LCD is a uniform representation and represents the schematic structure of the data. It is also a cycle directed graph and would arise when a data structure refers to itself recursively in its schema or data definition of a database (col. 13, lines 18-67), it is a logical graph and logical dependencies (see fig. 4). The dependency graph is represented for the data structure based on the database schema or data dictionary (col. 20, lines 46-67, col. 21, lines 1-67; and col. 39, lines 40-62). Thus, Applicants' arguments are not persuasive over the record of references.

2. Claims 1-8 and 30 have been cancelled.

3. Claims 9-29 are pending in this application.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,970,490 issued to Morgenstern in view of US Patent No. 5,493,682 issued to Tyra et al. (hereinafter Tyra).

With respect to claim 9, Morgenstern discloses querying a data catalog for direct dependencies of a code object and then for each dependency found (see fig. 4, col. 20, lines 46-67 and col. 21, lines 28-67; database schema is a kind of database catalog in the object-oriented or relational database: col. 5, lines 26-48; the code object of the of the dependency graph or tree is generated based on the code files: col. 8, lines 53-57. For each node object of a dependency tree is generated based on the code files and it is recursively executed based on the code or schema: col. 13, lines 25-28 and col. 20, lines 46-60).

Morgenstern discloses database schema of object-oriented database from which a dependency tree is generated based on the code files for the node objects and using recursive algorithm based on the schema or definition. Mongenstern does not explicitly indicate doing the query recursively until all basic dependencies are generated into a dependency tree.

However, Tyra discloses the object dependency being recursively executed until an object with no dependencies is located (col. 6, lines 22-37 and lines 52-62 and col. 7, lines 5-14).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern with the teachings of Tyra so as to obtain a dependency tree by doing a recursively execution on

the object dependency (Morgenstern - col. 6, lines 22-38). This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object dependency until on dependencies found (Tyra - col. 6, lines 22-37) in the dependency analysis and recursive path analysis DBMS procedures environment.

6. Claims 10-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,970,490 issued to Morgenstern in view of US Patent No. 5,493,682 issued to Tyra et al. (hereinafter Tyra) and further in view of US Patent No. 5,325,531 issued to McKeeman et al. (hereinafter McKeeman).

With respect to claims 10-16, Morgenstern in view of Tyra discloses a method as discussed in claim 9.

Morgenstern and in combination Tyra discloses generating a dependency tree, but do not explicitly indicate a database code object debugging tool, a database code coverage tool, a database code object profiling tool, testing tool, to identify dependent objects that are invalid in the database, cyclic dependencies and a dependency graph presentation tool.

However, McKeeman discloses debugging tools as debugger (see col. 5, lines 18-67 and col. 6, lines 1-12) and testing tool; invalid entries for a database, dependencies among database code object; and a dependency graph tools (col. 2, lines

32-44; col. 6, lines 25-45; col. 5, lines 18-67 and col. 6, lines 1-12; tools for software development: col. 6, lines 1-12).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern in view of Tyra with the teachings of Mckeeman so as to have debugging tools, testing tool for a dependency tree and a dependency graph presentation tool (col. 6, lines 1-12). The dependency graphs or trees for a recursive query of a database. This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object dependency until on dependencies found (Tyra - col. 6, lines 22-37) and having an optimizing compiler to produce object code from debugged source code modules (McKeeman - col. 6, lines 1-14) in the dependency analysis and recursive path analysis DBMS procedures environment.

7. Claim 17, 22 and 27-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,970,490 issued to Morgenstern in view of US Patent No. 5,493,682 issued to Tyra et al. (hereinafter Tyra) and further in view of US Patent No. 5,926,819 issued to Doo et al. (hereinafter Doo).

With respect to claim 17, Morgenstern discloses using a recursive algorithm for querying a database catalog for direct dependencies of a code object and then for each dependency found (see fig. 4, col. 20, lines 46-67 and col. 21, lines 28-67; database schema is a kind of database catalog in the object-oriented or relational database: col.

5, lines 26-48; the code object of the of the dependency graph or tree is generated based on the code files: col. 8, lines 53-57. For each node object of a dependency tree is generated based on the code files and it is recursively executed based on the code or schema: col. 13, lines 25-28 and col. 20, lines 46-60); using a parser on each of code objects in the dependency graph (col. 8, lines 9-20, and lines 38-52; also see col. 11, lines 25-38); and repeating step 1-3 for incorporating dependencies on triggers and their dependencies until new dependencies are not added to the dependency graph (col. 16, lines 52-67 and col. 17, lines 1-38). Tyra discloses doing the query recursively until all basic dependencies are generated into a dependency graph (the object dependency being recursively executed until an object with no dependencies is located: col. 6, lines 22-37 and lines 52-62 and col. 7, lines 5-14).

Morgenstern and in combination Tyra discloses generating a dependency tree, but do not explicitly indicate DML statements that "fire" triggers so as to identify dependencies on triggers and the triggers to incorporate the dependencies of the triggers into the dependency graph.

However, Doo is discloses DML statement being applied to fire the triggers (col. 5, lines 22-42, also see figs, 2-4).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern in view of Tyra with the teachings of Doo so as to have DML statement applying to fire the triggers (col. 5, lines 22-42). This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object

dependency until on dependencies found (Tyra - col. 6, lines 22-37) and using the DML statements to analyze the tree or graph (Doo – col. 6, lines 35-49) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claim 22, Morgenstern discloses applying a recursive algorithm that queries a database dependency information and of a database code object (see fig. 4, col. 20, lines 46-67 and col. 21, lines 28-67; database schema is a kind of database catalog in the object-oriented or relational database: col. 5, lines 26-48; the code object of the of the dependency graph or tree is generated based on the code files: col. 8, lines 53-57. For each node object of a dependency tree is generated based on the code files and it is recursively executed based on the code or schema: col. 13, lines 25-28 and col. 20, lines 46-60); using a parser on each of code objects in the dependency graph (col. 8, lines 9-20, and lines 38-52; also see col. 11, lines 25-38); and repeating step 1-3 for incorporating dependencies on triggers and their dependencies until new dependencies are not added to the dependency graph (col. 16, lines 52-67 and col. 17, lines 1-38). Tyra discloses outputs a direct dependency graph (the object dependency being recursively executed until an object with on dependencies is found: col. 6, lines 22-37 and lines 52-62 and col. 7, lines 5-14).

Morgenstern and in combination Tyra discloses generating a dependency tree, but do not explicitly indicate dependencies on triggers.

However, Doo is discloses the triggers (col. 5, lines 22-42, also see figs, 2-4).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern in view of

Tyra with the teachings of Doo so as to have DML statement applying to fire the triggers (col. 5, lines 22-42). This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object dependency until on dependencies found (Tyra - col. 6, lines 22-37) and using the DML statements to analyze the tree or graph (Doo – col. 6, lines 35-49) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claim 27, Morgenstern disclose a digital computer (col. 30, lines 17-28, see fig. 1); a database server couple to the computer (col. 28, lines 24-39); a database couple to the database server having data stored therein, the data including object oriented code data objects (object-oriented database; col. 13, lines 58-67); specifications of packages, implementations of packages, specifications of types, implementations of types (col. 3, lines 8-32) and the dependency graph being a data structure and having entries to contain representations of depending code objects, of packages, implementations of packages, specifications of types, implementations of types, triggers and dependencies of triggers which are relevant to the target data base code object (col. 20, lines 46-60). Tyra discloses a code mechanism for generating a dependency graph (see fig. 5, col. 6, lines 18-38 and lines 52-62).

Morgenstern and in combination Tyra discloses generating a dependency tree, but do not explicitly indicate dependencies on triggers and dependency of triggers.

However, Doo is discloses the triggers (col. 5, lines 22-42, also see figs, 2-4).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern in view of

Tyra with the teachings of Doo so as to have DML statement applying to fire the triggers (col. 5, lines 22-42). This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object dependency until on dependencies found (Tyra - col. 6, lines 22-37) and using the DML statements to analyze the tree or graph (Doo – col. 6, lines 35-49) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claim 28, Morgenstern disclose providing a database couple to the database server having data stored therein, the data including object oriented code data objects (object-oriented database; col. 13, lines 58-67); specifications of packages, implementations of packages, specifications of types, implementations of types (col. 3, lines 8-32) and the dependency graph being a data structure and having entries to contain representations of depending code objects, of packages, implementations of packages, specifications of types, implementations of types, triggers and dependencies of triggers which are relevant to the target data base code object (col. 20, lines 46-60). Tyra discloses using a recursive code mechanism for generating a dependency graph (see fig. 5, col. 6, lines 18-38 and lines 52-62).

Morgenstern and in combination Tyra discloses generating a dependency tree, but do not explicitly indicate dependencies on triggers and dependency of triggers.

However, Doo is discloses the triggers (col. 5, lines 22-42, also see figs, 2-4).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern in view of Tyra with the teachings of Doo so as to have DML statement applying to fire the triggers

(col. 5, lines 22-42). This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object dependency until on dependencies found (Tyra - col. 6, lines 22-37) and using the DML statements to analyze the tree or graph (Doo – col. 6, lines 35-49) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claim 29, Morgenstern disclose the target database code object the dependency graph being a data structure and having entries to contain representations of depending code object (col. 8, lines 38-67; object-oriented database; col. 13, lines 58-67); specifications of packages, implementations of packages, which are relevant to the target database code object (col. 3, lines 8-32 and col. 20, lines 46-60). Tyra discloses a recursive code mechanism for generating a dependency graph and a program code mechanism for using the dependency graph to debug the target database code object (see fig. 5, col. 6, lines 18-38 and lines 52-62).

Morgenstern and in combination Tyra discloses generating a dependency tree, but do not explicitly indicate dependencies on triggers and dependency of triggers.

However, Doo is discloses the triggers (col. 5, lines 22-42, also see figs, 2-4).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern in view of Tyra with the teachings of Doo so as to have DML statement applying to fire the triggers (col. 5, lines 22-42). This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object dependency until on dependencies found (Tyra - col. 6, lines 22-37) and using the DML

statements to analyze the tree or graph (Doo – col. 6, lines 35-49) in the dependency analysis and recursive path analysis DBMS procedures environment.

8. Claims 18-21 and 23-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,970,490 issued to Morgenstern in view of US Patent No. 5,493,682 issued to Tyra et al. (hereinafter Tyra) and further in view of US Patent No. 5,926,819 issued to Doo et al. (hereinafter Doo) and in view of US Patent No. 5,325,531 issued to McKeeman et al. (hereinafter McKeeman).

With respect to claims 18-21, Morgenstern in view of Tyra and Doo discloses a method as discussed in claim 17.

Morgenstern and in combination Tyra and Doo discloses generating a dependency tree and DML statements to fire the triggers, but do not explicitly indicate a database code object debugging tool, a database code coverage tool, a database code object profiling tool, testing tool, to identify dependent objects that are invalid in the database, cyclic dependencies and a dependency graph presentation tool.

However, McKeeman discloses debugging tools as debugger (see col. 5, lines 18-67 and col. 6, lines 1-12) and testing tool; invalid entries for a database, dependencies among database code object; and a dependency graph tools (col. 2, lines 32-44; col. 6, lines 25-45; col. 5, lines 18-67 and col. 6, lines 1-12; tools for software development: col. 6, lines 1-12).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern in view of Tyra and Doo with the teachings of Mckeeman so as to have debugging tools, testing tool for a dependency tree and a dependency graph presentation tool (col. 6, lines 1-12). The dependency graphs or trees for a recursive query of a database. This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object dependency until on dependencies found (Tyra - col. 6, lines 22-37) and having an optimizing compiler to produce object code from debugged source code modules (McKeeman - col. 6, lines 1-14) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claims 23-26, Morgenstern in view of Tyra and Doo discloses a method as discussed in claim 22.

Morgenstern and in combination Tyra and Doo discloses generating a dependency tree and DML statement to fire the triggers, but do not explicitly indicate a database code object debugging tool, a database code coverage tool, a database code object profiling tool, testing tool, to identify dependent objects that are invalid in the database, cyclic dependencies and a dependency graph presentation tool.

However, McKeeman discloses debugging tools as debugger (see col. 5, lines 18-67 and col. 6, lines 1-12) and testing tool; invalid entries for a database, dependencies among database code object; and a dependency graph tools (col. 2, lines 32-44; col. 6, lines 25-45; col. 5, lines 18-67 and col. 6, lines 1-12; tools for software development: col. 6, lines 1-12).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Morgenstern in view of Tyra and Doo with the teachings of Mckeeman so as to have debugging tools, testing tool for a dependency tree and a dependency graph presentation tool (col. 6, lines 1-12). The dependency graphs or trees for a recursive query of a database. This combination would made the method of generating a dependency tree based on code files (col. 8, lines 52-57) and recursively the object dependency until on dependencies found (Tyra - col. 6, lines 22-37) and having an optimizing compiler to produce object code from debugged source code modules (McKeeman - col. 6, lines 1-14) in the dependency analysis and recursive path analysis DBMS procedures environment.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

10. Any inquiry concerning this communication should be directed to Anh Ly whose telephone number is (703) 306-4527 via E-Mail: **ANH.LY@USPTO.GOV**. The examiner can be reached on Monday - Friday from 8:00 AM to 4:00 PM.

If attempts to reach the examiner are unsuccessful, see the examiner's supervisor, John Breen, can be reached on (703) 305-9790.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to: Central Office (703) 872-9306 (Central Official Fax Number)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA, Fourth Floor (receptionist).

Inquiries of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

ANH LY
MAY 14th, 2004

JEANNA M. CORRIELUS
PRIMARY EXAMINER